Predictive 3D search algorithm for multi-frame motion estimation
Lim, Hong Yin; Kassim, A.A.; de With, P.H.N.

Published in:
IEEE Transactions on Consumer Electronics

DOI:
10.1109/TCE.2008.4711256

Published: 01/01/2008

Document Version
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Abstract — Multi-frame motion estimation introduced in recent video standards such as H.264/AVC, helps to improve the rate-distortion performance and hence the video quality. This, however, comes at the expense of having a much higher computational complexity. In multi-frame motion estimation, there exists strong temporal correlation between the reference frames, which is not efficiently exploited in single-frame block-matching algorithms. In this paper, we propose a 3D motion search scheme which exploits the temporal correlation by using new 3D search patterns and motion vector predictors to obtain more accurate search centers. Compared to full search, our proposed algorithm results in PSNR losses of within 0.2 dB, while achieving a significantly lower motion estimation time by at least 96%. Furthermore, our results show that the proposed scheme is also significantly better than existing fast motion estimation algorithms for high-motion sequences 1.

Index Terms — multi-frame motion estimation, 3D motion vector predictors, 3D search pattern, hexagon pattern.

I. INTRODUCTION

Block matching motion estimation and compensation is used in video compression schemes to remove the inter-frame redundancy existing within video frames. In block matching motion estimation, a frame is partitioned into non-overlapping blocks; where the target block in the current frame is compared with candidate blocks in the reference frame within a search window. The best-matched block is obtained by minimizing a matching criterion/cost function, such as the Sum of Absolute Difference (SAD) or the Rate-Distortion Optimization (RDO) [1] used in the current H.264/AVC video standard [2]. The displacement between the current block position and that of the best-matched block in the reference frame is then represented as the motion vector (MV).

Multiple reference frame motion estimation (MRFME) [2] is one of the newest and most significant features introduced in the H.264/AVC video standard. In MRFME, the block-matching is conducted using several reference frames (maximum of 32). Unlike the conventional single-frame motion estimation used in older video coding standards such as MPEG-4, the MRFME is able to obtain better-matched blocks due to repetitive motion [3] and also able to overcome problems due to occlusion, different shadowing and lighting conditions.

The exhaustive search method (i.e., full search) for motion estimation, which involves searching all possible displaced locations within a search area, significantly increases the computational complexity of video encoders. Fast motion estimation (FME) algorithms evaluate fewer and only more probable locations and thus are computationally more efficient. Well-known FME algorithms such as the Diamond-Based Search (DS) [4] and Hexagon-Based Search (HEXBS) [5] are examples of center-biased algorithms, which perform the motion search centered on the origin of the search window. More recent algorithms such as UMHexagonS [6],[7] also use motion vector predictors to predict the best MV and to obtain a more accurate search center for further refinement search.

Most FME algorithms are designed for single-frame motion estimation, where the motion estimation is performed in a frame-by-frame way and each reference frame is searched evenly [8]. Thus, conventional single-frame motion estimation algorithms are unable to exploit the temporal correlation and motion characteristics [8] between multiple reference frames as in MRFME algorithms.

In this paper, we propose a new algorithm that extends current search schemes into a 3D search strategy, by using 3D motion vector (MV) predictors and 3D search patterns. Furthermore, the search strategies are able to deal with high-motion activity blocks and simplify the search for low-motion activity blocks. The rest of the paper is organized as follows. Section II provides an overview of current FME algorithms. Section III introduces the proposed 3D search patterns while the proposed 3D MV predictors are explained in Section IV. Our proposed strategy to deal with high-motion and low-motion activity blocks are presented in Sections V and VI while a summary of our algorithm is presented in Section VII. The algorithm’s performance in comparison with other FME algorithms is presented in Section VIII. We provide our concluding remarks in Section IX.

II. OVERVIEW OF FME ALGORITHMS

In FME algorithms [9], the block-matching cost function increases monotonically as the search moves away from the position of the best match/global minimum. The search will
thus converge towards the global minimum (optimal MV) if it is performed in the direction of the greatest decrease in the cost function. However, there is also a risk of the search getting trapped in a local minima, and thus failing to find the optimal MV. As a result, the same rate-distortion performance as the full search (FS) cannot be achieved. However, the trade-off between the quality of the match and the number of evaluations (a drastic reduction in search points compared to FS) is usually good.

The approach used by FME algorithms can basically be characterized by the following steps:

1. Evaluate the cost function at a few possible points; usually at points referenced by search patterns (i.e., Diamond [4] and Hexagon [5] search patterns).
2. Compare the resultant costs and re-centre the search in the position of the best point.
3. Repeat Steps 1 and 2 for either a fixed number of iterations, or until the algorithm converges on a local minimum.

In more recent FME algorithms such as UMHexagonS, a set of candidate MV predictors (i.e. MVs of spatially and temporally adjacent blocks [10]) are initially evaluated in order to estimate the best MV. In these predictive FME algorithms, the predictor with the minimum cost function (best predictor) is identified from all the possible candidate MV predictors. An early-termination criteria is then evaluated to determine whether to terminate the search. This criteria is usually in the form of a threshold which is compared against the minimum cost function. If the cost is less than the threshold, the best predictor is determined as the best MV and the MV search is terminated. Otherwise, the best predictor forms the search center (non center-biased) for a more detailed search using the various search patterns, as explained in Steps 1-3.

III. 3D SEARCH PATTERN SCHEME

Performing a search across frames using a 3D search pattern [8] provides an effective and direct way to locate the minimum distortion point (i.e., least RDO cost function [2]) in nearby frames. In a 3D search, the search can proceed directly to points at nearby frames and there is no necessity to perform a new search from the window center for each reference frame, as in conventional single frame FME algorithms (see Fig. 1). As a result, the computational time can be reduced.

We introduce the 3D-hexagon (3D-HEX) search pattern modeled on the 2D hexagon pattern which has been shown [5] to be able to locate the best MV as effectively as the Diamond pattern, but with a lower number of search points. In addition to the new 3D search pattern, we also introduce a multi-directional hexagon search pattern; where each of the four hexagonal patterns covers the horizontal, vertical, diagonal and anti-diagonal directions respectively (see Fig. 2).

A. Multi-Directional Hexagon Search Pattern

The concept of the multi-directional hexagon search pattern is similar to the concept of the directional search patterns of [11],[12]. The use of a directional search pattern enables a more accurate minimum point to be found with less search points. In the directional search, a suitable directional pattern is used based on the direction of the minimum point from the search pattern center.

Consider the example shown in Fig. 3, where the Large Diamond Pattern (LDP – indicated by ‘1’) is initially used since the most suitable direction is not known at first. After evaluating the points in the LDP, a suitable directional
hexagon search pattern is then used, based on the direction of the best/minimum point in LDP. After performing the initial search using the LDP, only a fixed hexagon pattern [5] is used subsequently regardless of the direction of the best point in Fig. 3(a), while in Fig. 3(b) a multi-directional hexagon pattern (Fig. 2) is used subsequently. The search is continuously conducted, using the fixed hexagon pattern in Fig. 3(a) and multi-directional hexagon pattern in Fig. 3(b), until the best point remains at the center of the hexagon pattern. Next, in the final refinement search, the Small Hexagon pattern (or Small Diamond pattern – “4”) [5] is used until the best point is at the pattern center at which point, the best MV is obtained.

In the example of Fig. 3, the initial best point (indicated by the first arrow) obtained using the LDP lies in the ‘Diagonal’ position (see Fig. 2c); hence the use of the Diagonal Hexagon pattern for the next search step (“2”) in Fig. 3(b). In this search step (“2”), since the best point still lies in the ‘Diagonal’ direction, the Diagonal Hexagon pattern is still used for the next search step (“3”). In this search step, the best point is found to be at the center of the hexagon pattern and so the final refinement search takes place next using the Small Hexagon pattern repeatedly (steps “4” & “5”) until the best MV is obtained.

As shown in Fig. 3, the number of search points used to reach the optimal MV (-3,4) for the fixed and multi-directional hexagon pattern are 24 and 21 points respectively, which shows that the multi-directional hexagon pattern results in a lower number of search points.

An experiment was conducted to study the general direction of the next best position in relation to that of the current best position. In the experiment, a LDP search centered on the best predictor from a set of candidates MV predictors used in UMHexagonS [10] is performed. The best point (i.e., with minimum cost function) in the LDP, T1 and its corresponding direction (Fig. 2) is noted. All points within a range of ± 2 from T1 are then evaluated and the direction of the best point from T1 (Next Best Position) is recorded. The results obtained in this experiment, averaged for different test sequences and encoded at different frame-rates, are shown in Table I.

<table>
<thead>
<tr>
<th>Best Position in LDP, T1</th>
<th>Next Best Position (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>Horizontal</td>
</tr>
<tr>
<td>Center</td>
<td>100.0</td>
</tr>
<tr>
<td>Horizontal</td>
<td>12.4</td>
</tr>
<tr>
<td>Vertical</td>
<td>8.7</td>
</tr>
<tr>
<td>Diagonal</td>
<td>15.4</td>
</tr>
<tr>
<td>Anti-Diagonal</td>
<td>12.1</td>
</tr>
</tbody>
</table>

As seen in Table I, there exists strong correlation between the direction of the next best position and the best position obtained from the LDP search (T1). From the results shown, when the best position obtained in LDP is in the ‘Horizontal’ direction; the next best position is also most likely in the ‘Horizontal’ direction (> 50%). Similarly, when the best position obtained in LDP is in the ‘Anti-Diagonal’ direction; the next best position is also most
likely obtained in the ‘Anti-Diagonal’ direction (> 45%). Therefore, it makes sense to design a multi-directional search pattern, where the appropriate directional pattern is determined based on the best position of the previous search pattern.

B. 3D Search Patterns

In this paper, we propose 3D Multi-Directional Hexagon Search pattern (3D MD-HEXS) and the 3D Diamond Search pattern which are illustrated in Fig. 4 and Fig 5, respectively. In 3D MD-HEXS, the Thick Hexagon pattern [5], [13] is applied on the current/best reference frame (ref \( t \)), where the center of the 3D pattern (3D search center) is located. For the nearest references (ref \( t \pm 1 \)), the Flatted Hexagon pattern [14] is applied while the Small Hexagon [5] pattern is used for the remaining references. The 3D Diamond Search pattern consists of the 3D-Large Diamond pattern (3D-LDP) and 3D-Small Diamond pattern (3D-SDP).

\( D = D_0 \times P_i \)

where \( P_i \) is the center position ‘0’ at reference \( i (i= t\pm1, t\pm2, \ldots t\pm n) \); \( D_0 \) is the location of the 3D search center (position ‘0’ at ref \( t \)); \( D_i \) is the temporal distance of the reference \( i \) from the frame being encoded; and \( D_0 \) is the temporal distance of ref \( t \) from the frame being encoded. This is illustrated in Fig. 6.
a. 3D-Large Diamond pattern (3D-LDP).

b. 3D-Small Diamond pattern (3D-SDP).

C. Proposed 3D Search Scheme

The 3D-LDP is used for the initial search in our 3D search scheme. The position with the minimum cost function, $J_{\text{min}}$ in the 3D-LDP (points 0 to 8 in Fig. 5a) is then used to select one of the four directional hexagon search patterns as shown in Table II. The points on the 3D hexagonal search pattern are then evaluated and the resulting position with $J_{\text{min}}$ is then used to select the directional search pattern for the next search step. This process is repeated until $J_{\text{min}}$ lies in the 3D search center, at which point, the search is concluded by using the 3D-SDP. Our 3D search scheme is summarized in Fig. 7.

![Fig. 5. 3D-Diamond Search pattern.](image)

![Fig. 6. Continuity of motion across video frames.](image)

![Fig. 7. Illustration of our 3D search scheme.](image)

### IV. 3D Motion Vector Predictors

The use of MV predictors in predictive algorithms [6],[10] significantly reduces the computational time while retaining the video quality. These MV predictors are selected based on spatial and temporal correlations between the current block and its adjacent blocks. In our 3D search algorithm, the MV is based on the 3D (x, y, t) coordinate system, where the third coordinate t refers to the reference frame index. Therefore, our candidate MV predictors are evaluated using the MV direction, (x, y) and the reference frame is located in (t). As in other predictive algorithms, the candidate MV predictors are initially evaluated and the best predictor later forms the 3D search center for the 3D search scheme defined in Section III.

The candidate MV predictors used in our 3D search algorithm are as follows:

i. Spatial – MV of the spatially adjacent blocks: left, top, top-left, top-right [10] (Fig. 8a); the stationary block- (0,0) in every reference; and the predicted MV [2].

ii. Temporal – MV of the co-located block and its adjacent blocks in the previous frame [10] (Fig. 8b).

iii. Up-layer Blocks – MVs of the upper-level block sizes are used as predictors for lower-level block sizes since there are strong correlation between the MVs of the different block sizes [6],[7] (Fig. 9). E.g., MV of block sizes 16×16, 8×16, and 16×8 are predictors for MV of 8×8 block size.
Fig. 8. Candidates for spatial and temporal predictors.

Fig. 9. Candidates for up-layer block predictors.

Fig. 10. Percentage of the best 3D MV predictors within ±1 from the optimal MV.

Legend: Bus1-Bus@30fps, Bus2-Bus@15fps; Foot1-Football@30fps, Foot2-Football@15fps; Stef1-Stefan@30fps, Stef2-Stefan@15fps; Fore1-Foreman@30fps, Fore2-Foreman@15fps

V. MULTI-HEXAGON-GRID SEARCH FOR HIGH-MOTION

Most center-biased search algorithms such as DS and HEXBS perform poorly when used on high-motion sequences and when the search range is large [6]. In our experiments as shown in Figs. 13-16 which are rate-distortion plots for various high-motion video sequences, a large drop in the coding efficiency is noted when DS and HEXBS are used (e.g. Bus). This is because the search pattern used is too small, and thus not suitable for accurately searching large and complex motion. To overcome this problem, we propose the use of a large search pattern, specifically the Multi-Hexagon-Grid Search pattern [6] (Fig. 11) for high-motion activity blocks.

One possible way to determine the motion activity is through the use of the minimum cost functions (J_{min}) of spatially adjacent blocks. High-motion activity can be inferred when the J_{min} obtained from the 3D search is significantly higher than the J_{min} of adjacent blocks, as this would imply that the optimal MV for the current block is most likely significantly different from the best MV obtained from the 3D search.

Table III shows the ratio of J_{min} obtained from the 3D search in the current block M_1 over the minimum of the J_{min} from adjacent blocks M_2 obtained for high-motion sequences with:

\[ M_2 = \text{MIN} (J_1, J_2, J_3, J_4) \]  \hspace{1cm} (2)

where J_1, J_2, J_3, J_4 are the minimum cost function of the spatially adjacent blocks.

In Table III, the ratio is only recorded when the difference in magnitude between the best MV obtained from the refinement search and the optimal MV obtained from FS is greater than 5. Since the optimal MV is significantly different (in this case, the significant difference is defined to be >5) from the best MV found using the refinement search, this indicates that a larger search method, e.g. Multi-Hexagon-Grid Search should be used. As seen from Table III, when the are only evaluated once in the motion estimation process. Therefore, this results in a lower computational complexity.

V. MULTI-HEXAGON-GRID SEARCH FOR HIGH-MOTION

Most center-biased search algorithms such as DS and HEXBS perform poorly when used on high-motion sequences and when the search range is large [6]. In our experiments as shown in Figs. 13-16 which are rate-distortion plots for various high-motion video sequences, a large drop in the coding efficiency is noted when DS and HEXBS are used (e.g. Bus). This is because the search pattern used is too small, and thus not suitable for accurately searching large and complex motion. To overcome this problem, we propose the use of a large search pattern, specifically the Multi-Hexagon-Grid Search pattern [6] (Fig. 11) for high-motion activity blocks.

One possible way to determine the motion activity is through the use of the minimum cost functions (J_{min}) of spatially adjacent blocks. High-motion activity can be inferred when the J_{min} obtained from the 3D search is significantly higher than the J_{min} of adjacent blocks, as this would imply that the optimal MV for the current block is most likely significantly different from the best MV obtained from the 3D search.

Table III shows the ratio of J_{min} obtained from the 3D search in the current block M_1 over the minimum of the J_{min} from adjacent blocks M_2 obtained for high-motion sequences with:

\[ M_2 = \text{MIN} (J_1, J_2, J_3, J_4) \]  \hspace{1cm} (2)

where J_1, J_2, J_3, J_4 are the minimum cost function of the spatially adjacent blocks.

In Table III, the ratio is only recorded when the difference in magnitude between the best MV obtained from the refinement search and the optimal MV obtained from FS is greater than 5. Since the optimal MV is significantly different (in this case, the significant difference is defined to be >5) from the best MV found using the refinement search, this indicates that a larger search method, e.g. Multi-Hexagon-Grid Search should be used. As seen from Table III, when the
optimal MV obtained from FS differs significantly from the best MV obtained through the proposed 3D search, the ratio is significantly high: larger than 2.0.

### TABLE III
**RATIO OF CURRENT JMIN OVER JMIN OF ADJACENT BLOCKS WHEN CURRENT MV AND OPTIMAL MV FROM FS DIFFERS SIGNIFICANTLY**

<table>
<thead>
<tr>
<th>Sequence</th>
<th>FR (fps)</th>
<th>Ratio of Jmin</th>
<th>Sequence</th>
<th>FR (fps)</th>
<th>Ratio of Jmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus</td>
<td>30</td>
<td>2.22</td>
<td>Bus</td>
<td>15</td>
<td>2.43</td>
</tr>
<tr>
<td>Football</td>
<td>30</td>
<td>2.51</td>
<td>Football</td>
<td>15</td>
<td>2.49</td>
</tr>
<tr>
<td>Stefan</td>
<td>30</td>
<td>2.57</td>
<td>Stefan</td>
<td>15</td>
<td>2.84</td>
</tr>
</tbody>
</table>

Based on Table III, we use a threshold $T_H$ for determining the motion activity as follows:

$$T_H = \alpha \times \min (J_1, J_2, J_3, J_4)$$  \hfill (3)

where $\alpha$ is a fixed parameter and in our experiments, $\alpha$ is set at 2.0.

If the $J_{\text{min}}$ obtained from the 3D search is more than $T_H$, we use the Multi-Hexagon-Grid Search pattern to perform a large motion search. The center of the Multi-Hexagon-Grid Search pattern is aligned to the search window center of the best reference frame obtained from the 3D search. After evaluating all the points, the best MV is determined and another 3D search centered on this best MV is performed.

### VI. SIMPLIFYING SEARCH FOR LOW-MOTION BLOCKS

Table IV shows the percentage of the optimal MV obtained using FS, which lies within ±1 from the origin (%MV). For low-motion sequences such as Hall, Mother, Container, the results in Table IV show that more than 80% of the optimal MV is obtained within ±1 from the origin. Based on this and the fact that spatially adjacent blocks can effectively represent the local motion activity [10], we propose the following criterion to identify low-motion activity within a block:

If the spatially adjacent MVs is within ±1 from the origin, then the current block is determined to be of low-motion activity.

### TABLE IV
**RESULTS WHEN LOW-MOTION CRITERION IS IMPLEMENTED**

<table>
<thead>
<tr>
<th>Sequence</th>
<th>FR (fps)</th>
<th>% CR</th>
<th>% MV</th>
<th>Sequence</th>
<th>FR (fps)</th>
<th>% CR</th>
<th>% MV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hall</td>
<td>7.5</td>
<td>94.9</td>
<td>82.9</td>
<td>Bus</td>
<td>15</td>
<td>10.2</td>
<td>3.8</td>
</tr>
<tr>
<td>Hall</td>
<td>15</td>
<td>95.8</td>
<td>85.7</td>
<td>Bus</td>
<td>30</td>
<td>16.2</td>
<td>7.5</td>
</tr>
<tr>
<td>Mother</td>
<td>7.5</td>
<td>88.2</td>
<td>77.2</td>
<td>Football</td>
<td>15</td>
<td>24.7</td>
<td>12.4</td>
</tr>
<tr>
<td>Mother</td>
<td>15</td>
<td>99.1</td>
<td>82.4</td>
<td>Football</td>
<td>30</td>
<td>27.5</td>
<td>16.2</td>
</tr>
<tr>
<td>Container</td>
<td>7.5</td>
<td>93.6</td>
<td>78.9</td>
<td>Stefan</td>
<td>15</td>
<td>18.5</td>
<td>9.4</td>
</tr>
<tr>
<td>Container</td>
<td>15</td>
<td>94.6</td>
<td>80.9</td>
<td>Stefan</td>
<td>30</td>
<td>26.6</td>
<td>17.8</td>
</tr>
</tbody>
</table>

To examine the effectiveness of the proposed criterion, the percentage of blocks that are correctly identified as low-motion by the proposed criterion is also shown in Table IV (column marked "% CR"). For the low-motion sequences Hall, Mother and Container, a significant number of blocks are correctly identified (> 75%) to be of low-motion.

When the block is determined to be low-motion through the proposed criterion, the 3D search strategy in Section III.C is simplified to using only the 3D Small Diamond Search (Fig. 7).

### VII. SUMMARY OF PROPOSED ALGORITHM

Our proposed algorithm for the 3D search scheme is as follows:

1. **Step 1**: Evaluate all candidate 3D MV predictors (Section IV).
2. **Step 2**: If MV of spatially adjacent blocks lies within ±1 from origin, then block is of low-motion activity (Section VI).
3. **Step 3**: Perform 3D search (3D-LDP + 3D MD-HEXS + 3D-SDP), centered on the best MV predictor obtained from Step 1 (Section III). If block is low-motion, only perform the search using 3D-SDP.
4. **Step 4**: If $J_{\text{min}} > T_H$ determined from Eq.(3), then use Multi-Hexagon-Grid search pattern. Perform the 3D search, centered on the best MV (Section V).

### VIII. SIMULATION RESULTS

Our experiments were carried out using the JVT/H.264 reference software [15]. In our experiments, the H.264 main profile is used with RD Optimization and CABAC encoding. The sub-pixel motion estimation is disabled. The search range used is ±16 and ±32 for the QCIF and CIF sequences respectively and the number of reference frames used is 5. The measures used to compare the performance are:

a. $\delta$PSNR – Difference in Peak Signal-to-Noise ratio per frame, compared to full search (FS) measured in dB.

b. $\delta$Time – Percentage of savings over FS in terms of the ME time.

c. $\delta$Bitrate – Percentage of bitrate reduction.

Table V presents the average results obtained by encoding the test sequences with quantizer values of 24, 28, 32, 36 [16]. In particular, we compare our proposed 3D search algorithm (‘Proposed’) with Diamond Search (‘DS’), Hexagon-based Search (‘HEXS’), UMHexagonS (‘UMHexS’), and Recent-Biased Search using 3D Spiral Cross and Diamond pattern [8] (‘RBS’). From the results in Table V, it can be seen that the search conducted using 3D search patterns (RBS and our proposed algorithm) is the fastest; as shown by the lowest computational time ($\delta$Time).

Compared to FS, our method has a significant reduction in computational time (96-99%). Furthermore, the video quality does not degrade significantly as indicated by the decrease in PSNR of < 0.1 dB and the increase in encoding bitrate is marginal (less than 3%). At all bitrates, the drop in PSNR compared to FS is within 0.2 dB (see Figs. 12-15).

Compared to DS, HEXBS, UMHexagonS and RBS, our algorithm shows similar quality for the low-motion sequences (Hall, Mother, Container). For the high-motion
sequences such as Bus, Football, Canoa, and Stefan (Figs. 12-15), our algorithm is up to 0.5 dB better than DS and HEXBS, and up to 1 dB better than RBS. Compared to UMHexagonS, our algorithm is only 0.1 dB less. However, our method has a significant reduction in the motion estimation (ME) computations of up to 35% compared to DS and HEXBS, and 80% compared to UMHexagonS. Although our 3D search algorithm is slightly slower than RBS, it more than makes up for it with significantly better quality especially for the high-motion sequences. This is achieved through the use of our large search pattern (Multi-Hexagon-Grid Search pattern) and our 3D search pattern, which does a search along a continuous motion trajectory across the frames.

Overall, our algorithm is able to maintain the video quality while significantly reducing the computational time due to the effective exploitation of the temporal correlations between the multi-frames, by utilizing the 3D search patterns and 3D motion vector predictors.

### TABLE V

**PERFORMANCE OF FME ALGORITHMS COMPARED TO FS**

<table>
<thead>
<tr>
<th>Sequence</th>
<th>FR (fps)</th>
<th>Measures</th>
<th>DS</th>
<th>HEX-BS</th>
<th>UM-HexS</th>
<th>RBS</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hall (QCIF)</td>
<td>15</td>
<td>δPSNR</td>
<td>-0.01</td>
<td>-0.02</td>
<td>-0.01</td>
<td>-0.03</td>
<td>-0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δTime</td>
<td>94.8</td>
<td>95.5</td>
<td>94.3</td>
<td>96.0</td>
<td>96.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δBitrate</td>
<td>0.24</td>
<td>0.23</td>
<td>-0.17</td>
<td>0.01</td>
<td>-0.12</td>
</tr>
<tr>
<td>Mother (QCIF)</td>
<td>15</td>
<td>δPSNR</td>
<td>-0.01</td>
<td>-0.01</td>
<td>-0.01</td>
<td>-0.01</td>
<td>-0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δTime</td>
<td>94.8</td>
<td>95.3</td>
<td>92.7</td>
<td>96.1</td>
<td>96.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δBitrate</td>
<td>1.14</td>
<td>1.57</td>
<td>0.64</td>
<td>1.53</td>
<td>1.53</td>
</tr>
<tr>
<td>Container (QCIF)</td>
<td>15</td>
<td>δPSNR</td>
<td>-0.01</td>
<td>-0.01</td>
<td>-0.01</td>
<td>-0.01</td>
<td>-0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δTime</td>
<td>94.8</td>
<td>95.5</td>
<td>94.3</td>
<td>95.9</td>
<td>96.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δBitrate</td>
<td>-0.08</td>
<td>0.03</td>
<td>-0.11</td>
<td>1.19</td>
<td>0.21</td>
</tr>
<tr>
<td>Coastguard (QCIF)</td>
<td>30</td>
<td>δPSNR</td>
<td>-0.01</td>
<td>-0.01</td>
<td>-0.01</td>
<td>-0.02</td>
<td>-0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δTime</td>
<td>96.2</td>
<td>96.6</td>
<td>90.4</td>
<td>97.7</td>
<td>98.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δBitrate</td>
<td>0.36</td>
<td>0.49</td>
<td>0.15</td>
<td>1.44</td>
<td>1.07</td>
</tr>
<tr>
<td>Foreman (CIF)</td>
<td>30</td>
<td>δPSNR</td>
<td>-0.04</td>
<td>-0.06</td>
<td>-0.03</td>
<td>-0.06</td>
<td>-0.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δTime</td>
<td>98.4</td>
<td>98.7</td>
<td>96.7</td>
<td>99.2</td>
<td>99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δBitrate</td>
<td>1.96</td>
<td>3.68</td>
<td>0.49</td>
<td>4.73</td>
<td>2.46</td>
</tr>
<tr>
<td>Bus (CIF)</td>
<td>30</td>
<td>δPSNR</td>
<td>-0.03</td>
<td>-0.03</td>
<td>-0.01</td>
<td>-0.09</td>
<td>-0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δTime</td>
<td>98.5</td>
<td>98.7</td>
<td>96.1</td>
<td>99.3</td>
<td>99.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δBitrate</td>
<td>12.69</td>
<td>13.10</td>
<td>1.08</td>
<td>19.82</td>
<td>2.57</td>
</tr>
<tr>
<td>Football (CIF)</td>
<td>30</td>
<td>δPSNR</td>
<td>0.01</td>
<td>0.00</td>
<td>-0.01</td>
<td>0.02</td>
<td>-0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δTime</td>
<td>98.1</td>
<td>98.5</td>
<td>93.9</td>
<td>99.1</td>
<td>99.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δBitrate</td>
<td>3.17</td>
<td>3.39</td>
<td>0.64</td>
<td>5.02</td>
<td>1.14</td>
</tr>
<tr>
<td>Canoa (CIF)</td>
<td>30</td>
<td>δPSNR</td>
<td>0.00</td>
<td>0.01</td>
<td>0.00</td>
<td>-0.01</td>
<td>-0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δTime</td>
<td>98.4</td>
<td>98.7</td>
<td>94.8</td>
<td>99.3</td>
<td>99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δBitrate</td>
<td>2.29</td>
<td>2.64</td>
<td>0.34</td>
<td>4.86</td>
<td>0.94</td>
</tr>
<tr>
<td>Stefan (CIF)</td>
<td>30</td>
<td>δPSNR</td>
<td>-0.01</td>
<td>0.00</td>
<td>-0.01</td>
<td>0.00</td>
<td>-0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δTime</td>
<td>98.5</td>
<td>98.7</td>
<td>95.7%</td>
<td>99.2</td>
<td>99.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>δBitrate</td>
<td>9.84</td>
<td>9.78</td>
<td>0.65</td>
<td>22.42</td>
<td>2.97</td>
</tr>
</tbody>
</table>

**Fig. 12. Rate-Distortion plot for Bus @ 30fps.**

**Fig. 13. Rate-Distortion plot for Football @ 30fps.**

**Fig. 14. Rate-Distortion plot for Canoa @ 30fps.**
In this paper, we introduce a predictive 3D search pattern for multi-frame motion estimation. The 3D MV predictors are used to obtain a more accurate search center while the 3D search patterns are designed to track the motion trajectories along the reference frames. This reduces the computational complexity significantly while maintaining the search accuracy. In addition, we also implement a large motion complexity significantly while maintaining the search method utilizing the Multi-Hexagon Grid search pattern and saves on the search points for low-motion blocks by simplifying the 3D search. Our results show that our algorithm is better than the DS and HEXBS in terms of quality (e.g. high-motion sequences) and speed. Compared to the center-biased 3D search algorithm (RBS) proposed in [8], our algorithm has similar speed, but has significantly much better quality (more than 1 dB for high-motion sequences). On average, the PSNR loss for our algorithm is less than 0.2 dB while a savings of more than 96% in computational time is achieved compared to FS. This enables our algorithm to be a good candidate for real-time applications. For future research, we are looking into a 3D sub-pixel motion estimation scheme.

IX. CONCLUSION

REFERENCES


Lim Hong Yin studied electrical engineering at the National University of Singapore from 1999-2003. He is currently pursuing his Ph.D. under the supervision of Prof. Ashraf A. Kassim at the same University. His research interests include video coding, signal processing, and multimedia.

Ashraf A. Kassim received his B.Eng. (First Class Honors) in Electrical Engineering from the National University of Singapore (NUS) in 1985. From 1986 to 1998, he worked on machine vision systems at Texas Instruments. Ashraf went on to obtain his Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, in 1993. Since 1993, he has been with the Electrical & Computer Engineering Department at NUS, where he is currently the Vice-Dean. Dr. Kassim's research interests include image analysis, machine vision, video/image processing and compression.

Peter H.N. de With graduated in electrical engineering from the University of Technology in Eindhoven and received his Ph.D. degree from the University of Technology Delft, The Netherlands in 1992. He joined Philips Research Labs Eindhoven in 1984, where he became a member of the Magnetic Recording Systems Department. In 1996, he became senior TV systems architect and in 1997, he was appointed as full professor at the University of Mannheim, Germany, at the faculty Computer Engineering. In Mannheim he was heading the chair on Digital Circuitry and Simulation with the emphasis on video systems. Between 2000 and 2007, he was with LogicaCMG (now Logica) in Eindhoven as a principal consultant. Early 2008, he joined CycloMedia Technology, The Netherlands, as vice-president for video technology. Since 2000, he is professor at the University of Technology Eindhoven, at the faculty of Electrical Engineering and leading a chair on Video Coding and Architectures. Mr. de With is a Fellow of the IEEE, program committee member of the IEEE CES, ICIP and VCIP, board member of the IEEE Benelux Chapters for Information Theory and Consumer Electronics, co-editor of the historical book of this community, former scientific board member of LogicaCMG, scientific advisor to Philips Research, and of the Dutch Imaging school ASCII, IEEE ISCE and board member of various working groups.